Secure Middleware for Robust and Efficient Interoperability over Disadvantaged Grids

Dr. Ramesh Bharadwaj

Center for High Assurance Computer Systems
Naval Research Laboratory
Washington DC 20375 USA
Tel: +1-202-767-7210
Fax: +1-202-404-7942
Email: ramesh@itd.nrl.navy.mil

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 DEC 2007	**N/A**	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Secure Middleware for Robust and Efficient Interoperability over Disadvantaged Grids	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Center for High Assurance Computer Systems Naval Research Laboratory Washington DC 20375 USA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**45**	

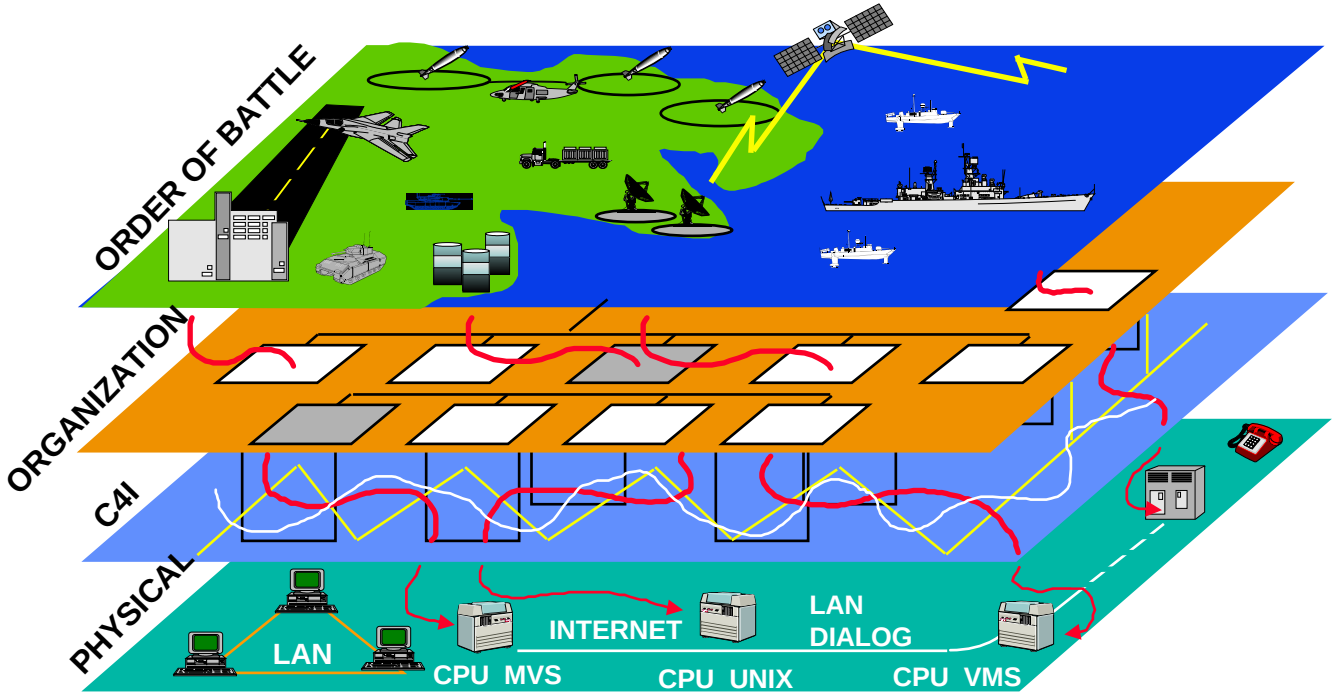
Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Roadmap

1. *Background and Motivation*
2. *Our Solution*
3. *Design Philosophy*
4. *Case Studies*
5. *Technical Approach*
6. *Major Accomplishments*
7. *Transition Plans*

1. Background and Motivation
2. Our Solution
3. Design Philosophy
4. Case Studies
5. Technical Approach
6. Major Accomplishments
7. Transition Plans

Network-Centric Warfare Demands a Secure and Survivable Information Grid



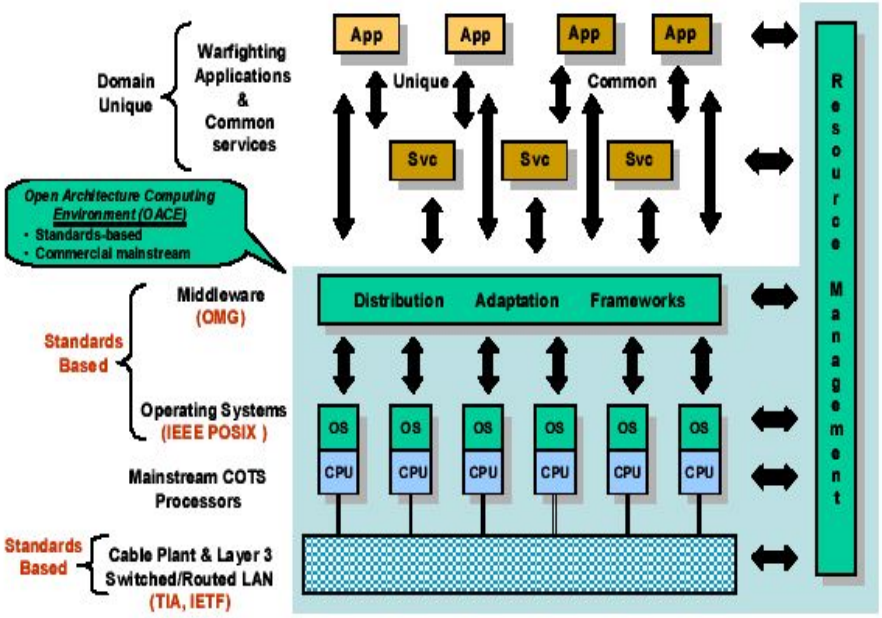
Information Ops Battle Space

Requirements for the Navy's Command and Information Infrastructure are flexibility, modular system design, fast and easy configuration, and information assurance.
-- Committee on Network-Centric Naval Forces

The Navy's Open Architecture: Requirements for Interoperability

" *[The Open Architecture will …]* **substantially reduce shipboard computer maintenance by capitalizing on the fact that application components are not bound to computer locality but instead are free to migrate to available processors under Resource Management (RM) control."**

Open Architecture Computing Environment (NSWC Dahlgren)



Infrastructure must provide:

- *Pool-of-computers architecture*
- *Applications not bound to computer locality but migrate to available processors*
- *Functionally distinct self-contained applications or components*
- *Components loosely coupled in space and time*
- *Applications built for portability and location transparent allocation and operation*

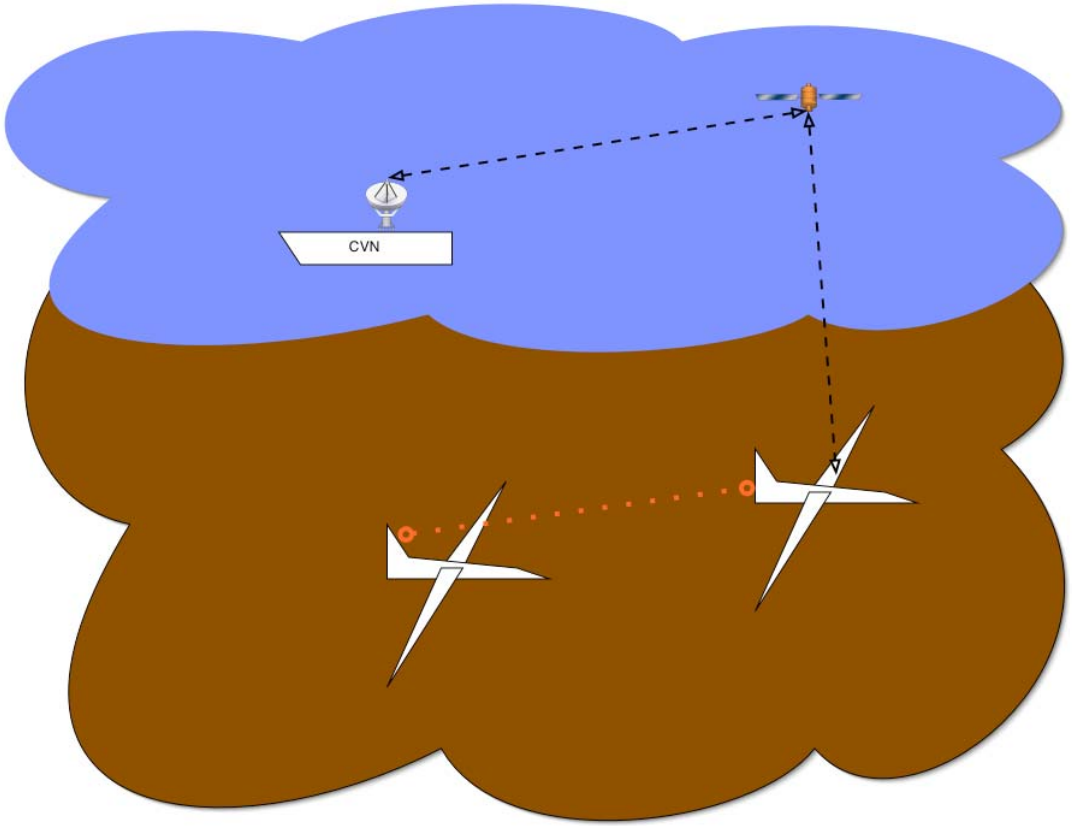
How can we achieve this?

Software agents are computer programs with one or more of the following attributes:

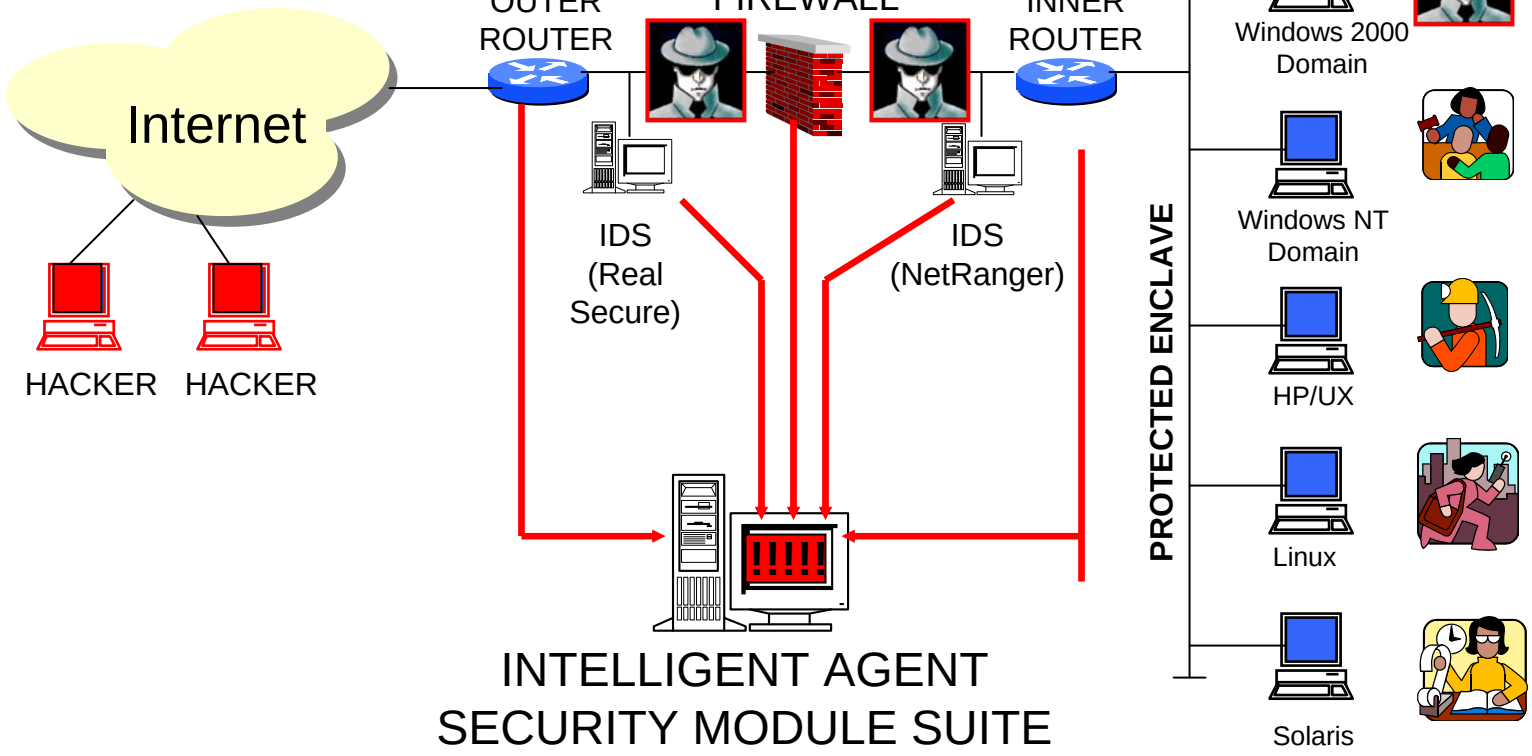
- *autonomy ("autonomous agents")*
- *mobility ("mobile agents")*
- *learned behavior ("learning agents")*

- *multiplicity ("multi-agent systems")*
- *distributed implementation*
- *cooperation and coordination*
- *"emergent" behavior*

A Case for Distributed Agents: UAV Swarms



A Case for Distributed Middleware:
Intelligent Agent Security Module



- Real-time Intrusion Pattern Detection
- Proactive Attack Identification
- Cyberlab – Effectiveness Metrics
- Identify Attack Sources
- Forensic Analysis and Data Mining
- Correlation, Fusion, and Visualization

Threats to Interoperability

"A Network Enabled Battlespace is dangerous if content is not secured and guaranteed. *[…]* a major challenge is to ensure that data and communications, at rest and on the fly, are secure each time, every time."

-- Battlespace Information 2003

Interoperability goals:

- *reduce total ownership costs*
- *quick and easy system upgrade and reconfiguration*
- *lower impact of COTS upgrades*
- *reduce compatibility problems*

THREATS

- COTS flaws
- Insiders
- Nation States
- Hackers
- User mistakes
- Trojan horses

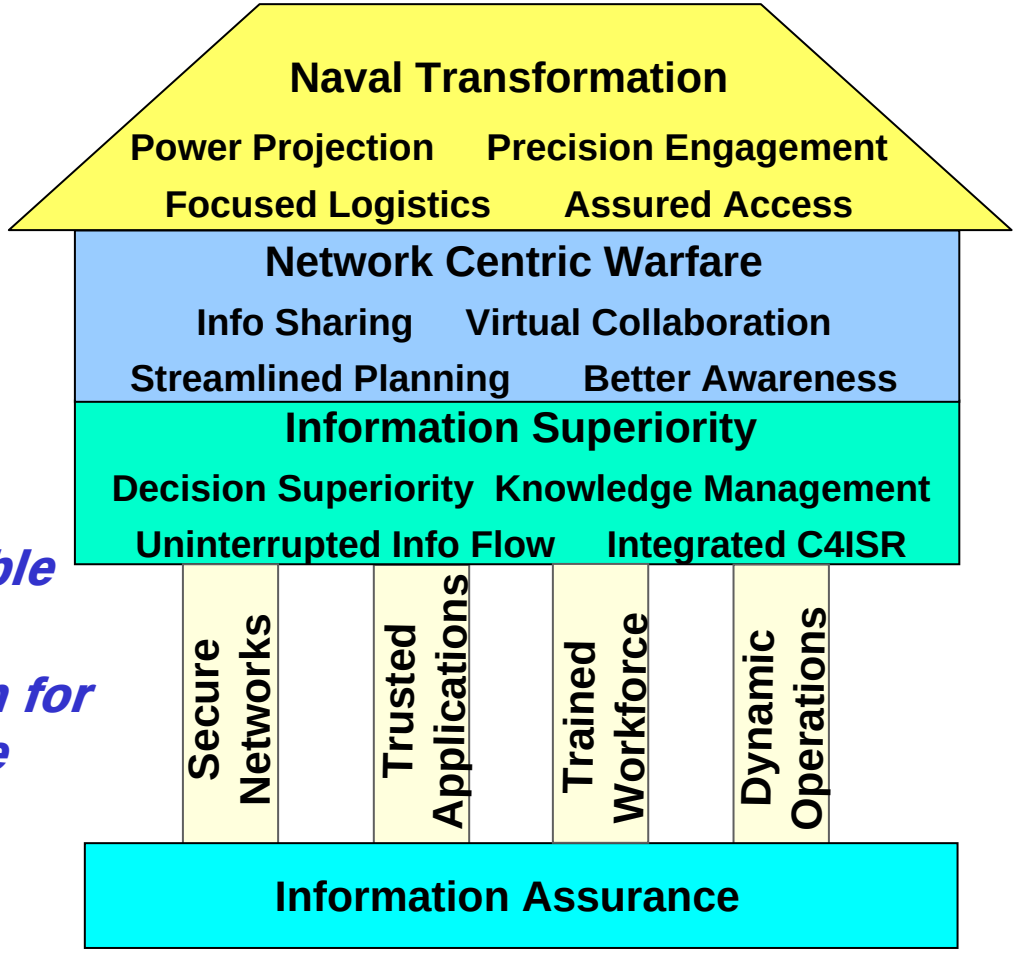
Information Assurance (IA)

"Information Operations That Protect and Defend Information and Information Systems by Ensuring Their Availability, Integrity, Authentication, Confidentiality, and Non-repudiation. This Includes Providing for Restoration of Information Systems by Incorporating Protection, Detection, and Reaction Capabilities."

Joint Doctrine for Information Operations
Joint Pub 3-13, Oct 9, 1998

IA Is An Enabler

- We ***Count*** on Information Superiority to Improve Combat Effectiveness
  - *Full Spectrum Dominance*
  - *Network Centric Warfare*
- IA ***Enables*** Information Superiority in a Network-Centric Paradigm
  - *Global Secure, Interoperable Network*
  - *State-of-the Art Protection for Information Infrastructure*



1. Background and Motivation
2. Our Solution
3. Design Philosophy
4. Case Studies
5. Technical Approach
6. Major Accomplishments
7. Transition Plans

Solution: Secure and Reconfigurable Middleware

Distributed middleware researchers[1] identify the following challenges:

- *Programming Abstractions*
- *Naming and Resource Discovery*
- *Adaptive Data Fusion*
- *Adaptive Distributed Plumbing*
- *Failure Semantics*
- *Runtime Mechanisms*
- *System Evaluation*

… but miss the most important[2] ones:

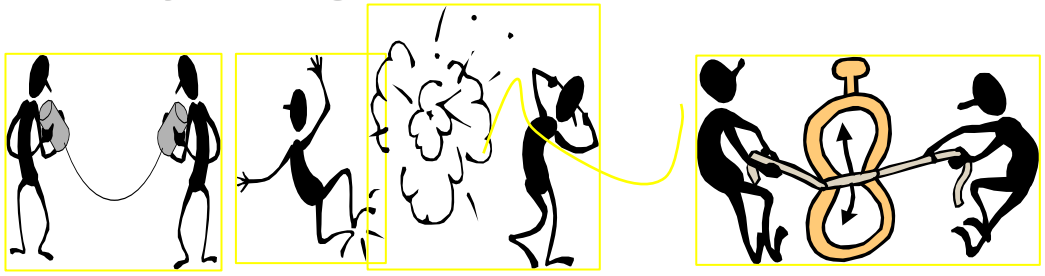
- *Trustworthiness*
- *Security*
- *Robustness*
- *System Survivability*

[1] Ramachandran U., et al., 9th IEEE Workshop on Future Trends of Distributed Computing Systems, May 2003.

[2] Bharadwaj R., 9th IEEE Workshop on Future Trends of Distributed Computing Systems, May 2003.

Secure Infrastructure for Networked Systems (SINS)

- Uses software agents technology
- Addresses security, performance, and robustness (survivability addressed in a related NRL 6.2 project)
- Builds security into agent middleware



What can we prove about agents in the SINS architecture?

- Completeness and Consistency of Agent Behavior
- Mechanical proofs of safety properties and agent compliance with local security policies
- Determination of emergent behavior of a community of agents

Security Agents Enforce a Consistent Security Policy

CRYPTO ASSIST AGENTS

SECURITY AGENTS

SAFETY PROPERTY
Never issue a CFF if forceCode == <friendly>

MONITORING AGENTS

APPLICATION-SPECIFIC AGENTS

- intrusion detection
- application monitoring
- survivability
- infrastructure monitoring

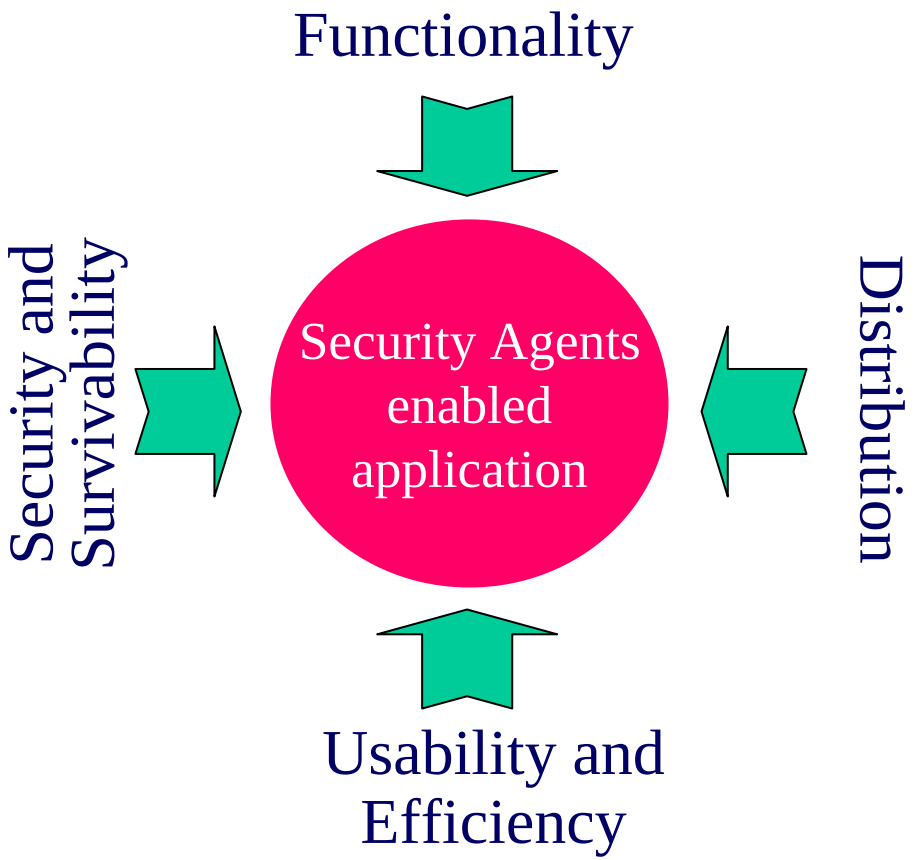
AUTHORIZATION AGENTS

POLICY ENFORCEMENT AGENTS

Security Agents act as mini-firewalls between an application and the OS resources.

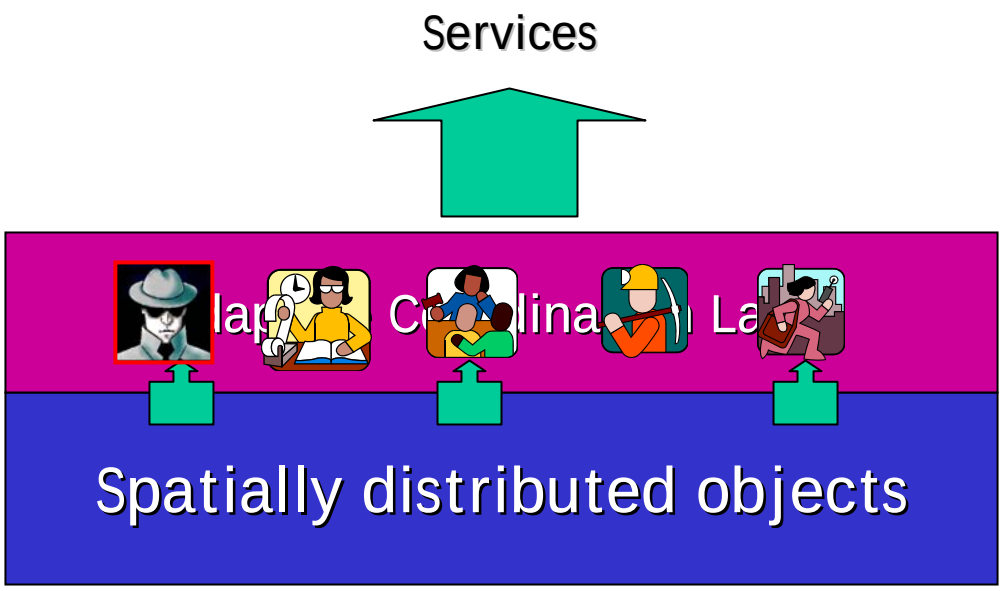
1. Background and Motivation
2. Our Solution
3. **Design Philosophy**
4. Case Studies
5. Technical Approach
6. Major Accomplishments
7. Transition Plans

Design Tradeoffs



Security and Survivability must be considered in the context of applications.

Based on a Dual-Layer Approach



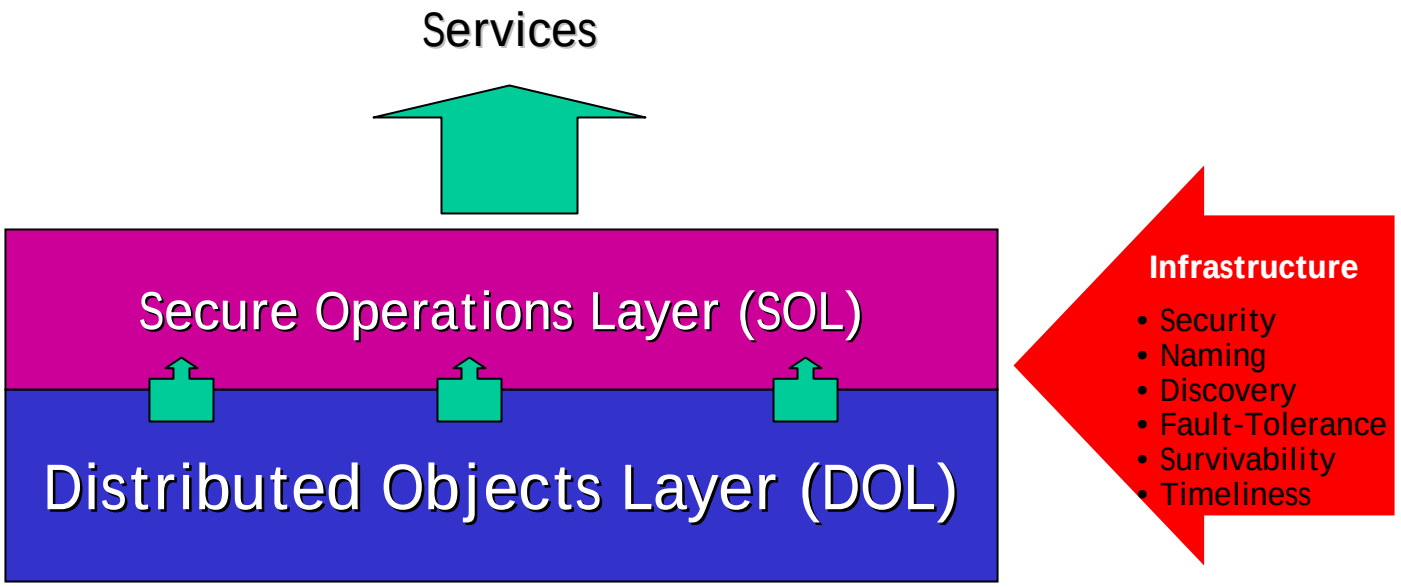
References:

• Bharadwaj R, “SOL: A Verifiable Synchronous Language for Reactive Systems,” In Proc. Synchronous Languages, Applications, and Programming (SLAP’02), ETAPS 2002, Grenoble, France, April 2002.

• Bharadwaj R, Froscher J, Khashnobish A and Tracy J. “An Infrastructure for Secure Interoperability of Agents,” in Proc. Sixth World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL July 2002.

• Bharadwaj R, “SINS: A Middleware for Autonomous Agents and Secure Code Mobility,” In Proc. Second International Workshop on Security of Mobile Multi-Agent Systems (SEMAS-02), First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), Bologna, Italy, July 2002.

Secure Infrastructure for Networked Systems (SINS)

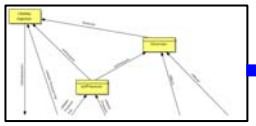


Domain Engineering: Identification and Design of SOL Components

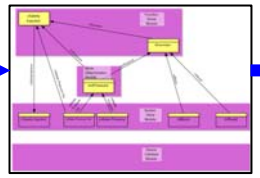
- Bharadwaj R. "Formal Analysis of Domain Models," in Proc. International Workshop on Requirements for High Assurance Systems (RHAS'02), Essen, Germany, September 2002.
- Kirby J. "Rewriting Requirements for Design," in Proc. IASTED International Conference on Software Engineering and Applications (SEA 2002), Cambridge MA, November 2002.
- Bharadwaj R. "How to fake a Rational Design Process using the SCR Method," in Proc. Software Engineering for High Assurance Systems (SEHAS 2003), held in conjunction with the International Conference on Software Engineering (ICSE), Portland OR, May 2003.

Secure Agent Development Process

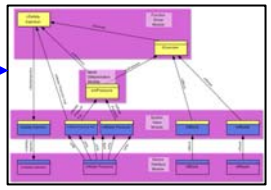
Secure Agent Requirements



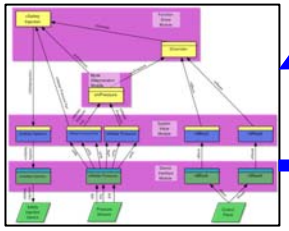
Standard Decomposition



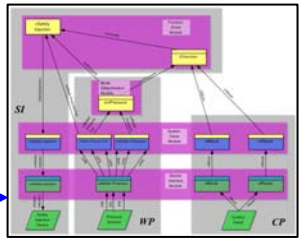
Agent Design



Agent Implementation



Agent Deployment



1. Background and Motivation
2. Our Solution
3. Design Philosophy
4. **Case Studies**
5. Technical Approach
6. Major Accomplishments
7. Transition Plans

Case Studies

Next-Generation agent-based Command and Control Systems:

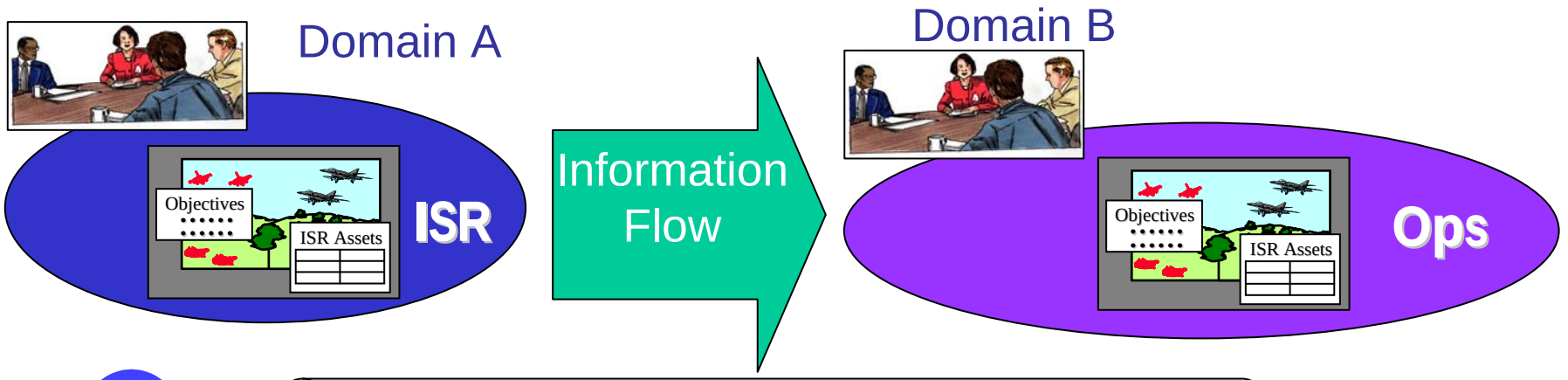
- Integrated Marine Multi-Agent Command and Control System (IMMACCS): Agent-based C2 system
- Real-time Execution Decision Support (REDS): Decision Support System which uses agents for information access and dissemination

Current agent-based systems *cannot guarantee:*

- Integrity: System safety and information assurance are not considered
- Performance: The distributed object model is inefficient
- Robustness : Agents are brittle, hard to create, deploy, and debug

Case Study: IMMACCS

System Integrity



Agent at Domain A

```
if Radar.forceCode == <not friendly> &&
   Radar.status == ACTIVE
then
    CallForFire.target = name (Radar)
    CallForFire.controlMethod = WHEN READY
endif
```

SADL

Integrity factors

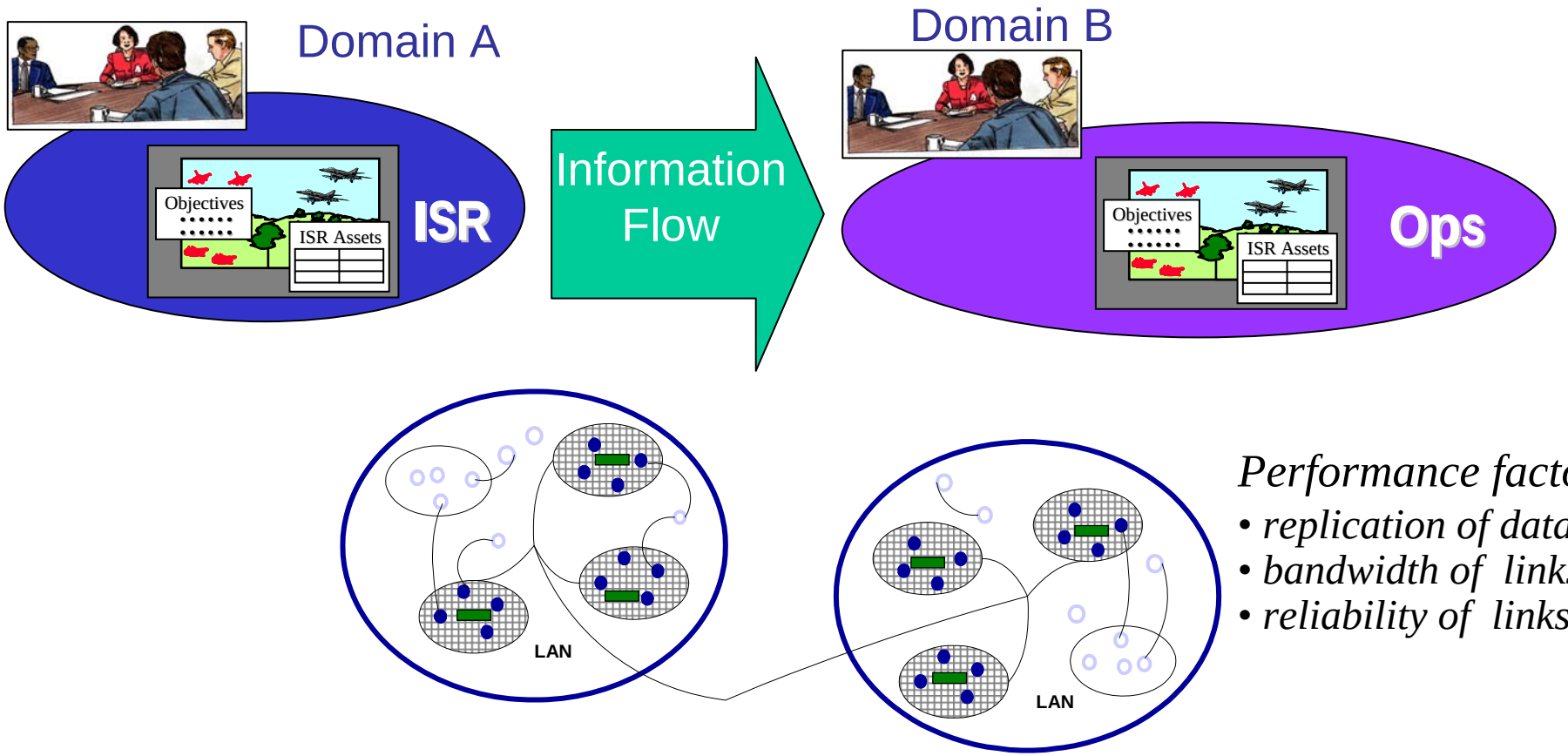
- *information leaks*
- *user mistakes*
- *malicious attacks*

Safety Property

Never issue a Call For Fire if forceCode == <friendly>

Case Study: IMMACCS

Performance



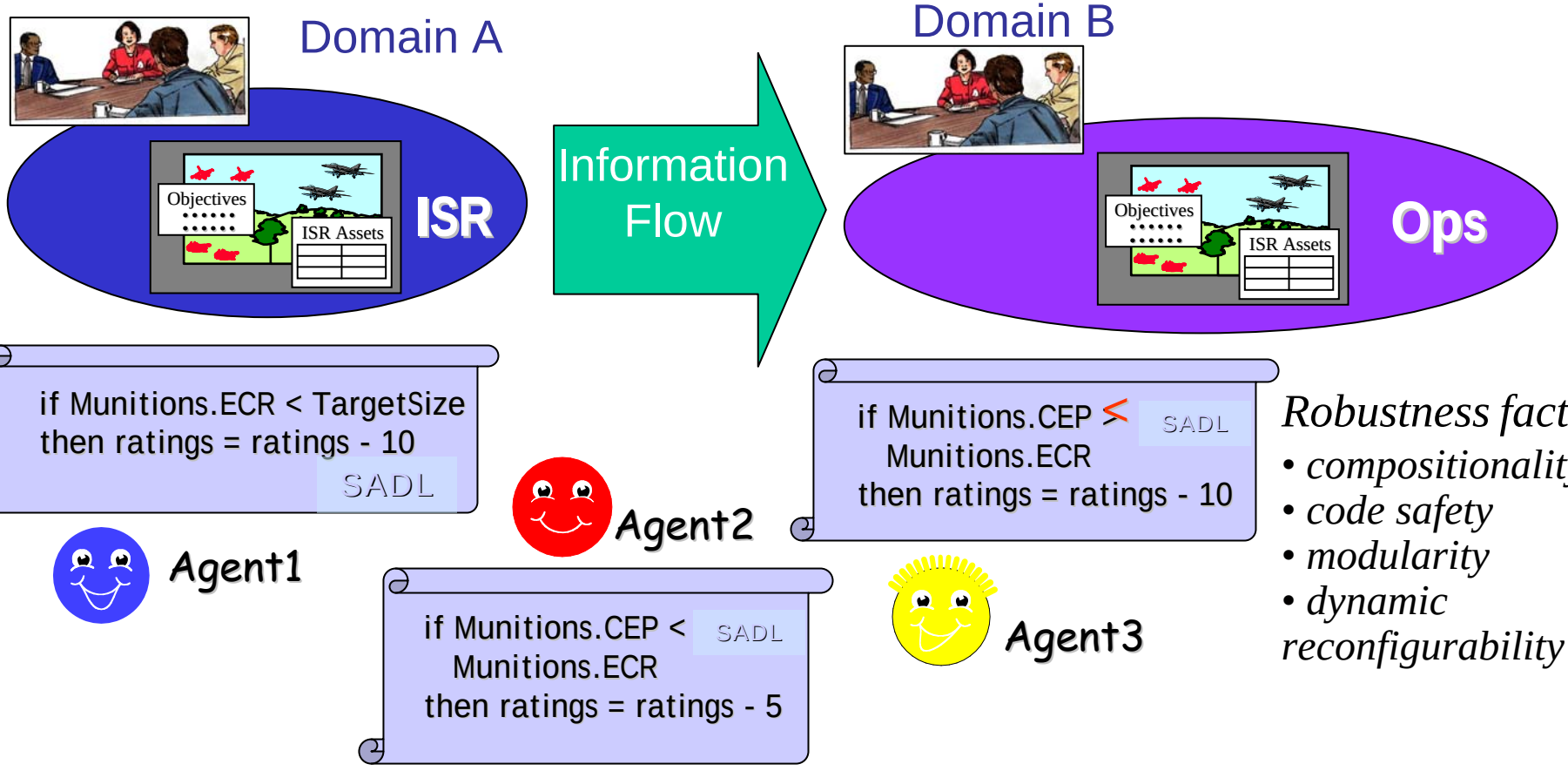
Performance factors

- *replication of data*
- *bandwidth of links*
- *reliability of links*

Distributed Objects

Case Study: IMMACCS

Robustness



Robustness factors

- *compositionality*
- *code safety*
- *modularity*
- *dynamic reconfigurability*

Evaluating agent behavior

Completeness and consistency of emergent agent behavior

1. Background and Motivation
2. Our Solution
3. Design Philosophy
4. Case Studies
5. **Technical Approach**
6. Major Accomplishments
7. Transition Plans

Three-Pronged Approach

SYSTEM INTEGRITY

- Authentication and authorization
- Confidentiality and integrity of transmitted information
- Security Protocols for fast/easy configuration
- Safety and Security Policy Enforcement

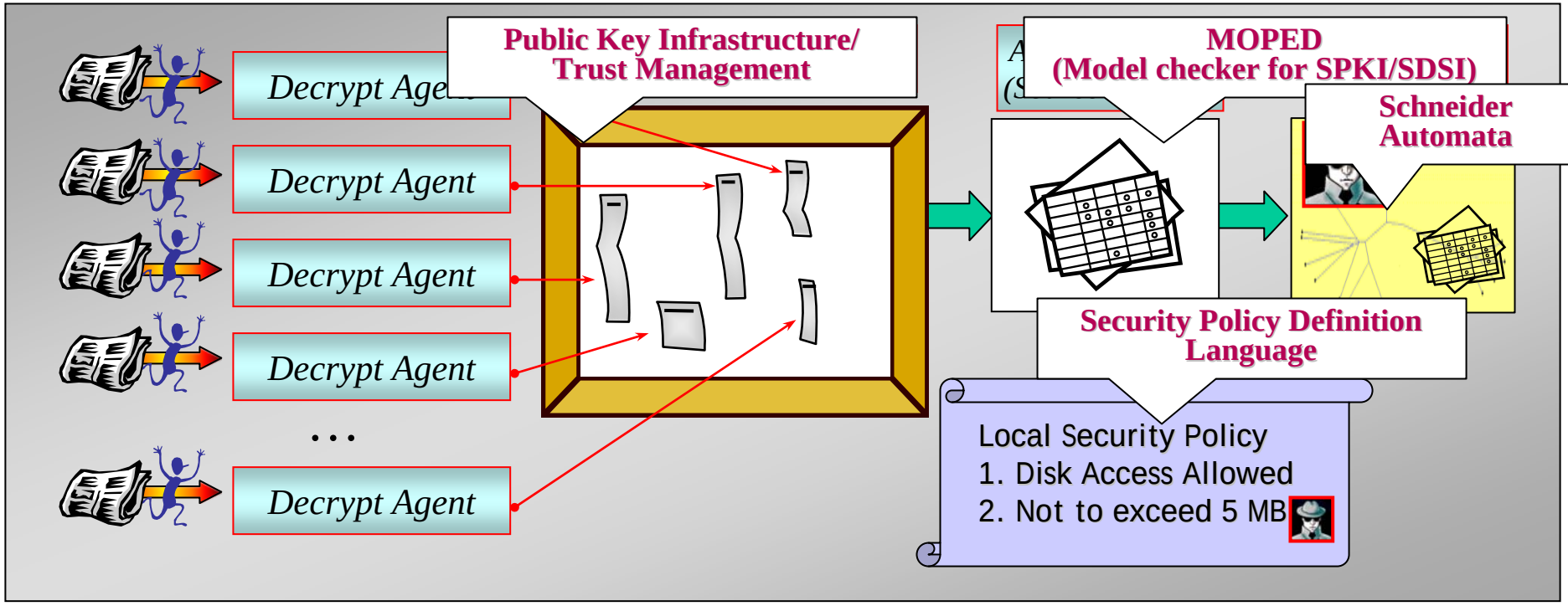
PERFORMANCE

- Dynamically determined agent routing patterns
- Flexible event handling and propagation
- Highly-efficient transmission of relevant information

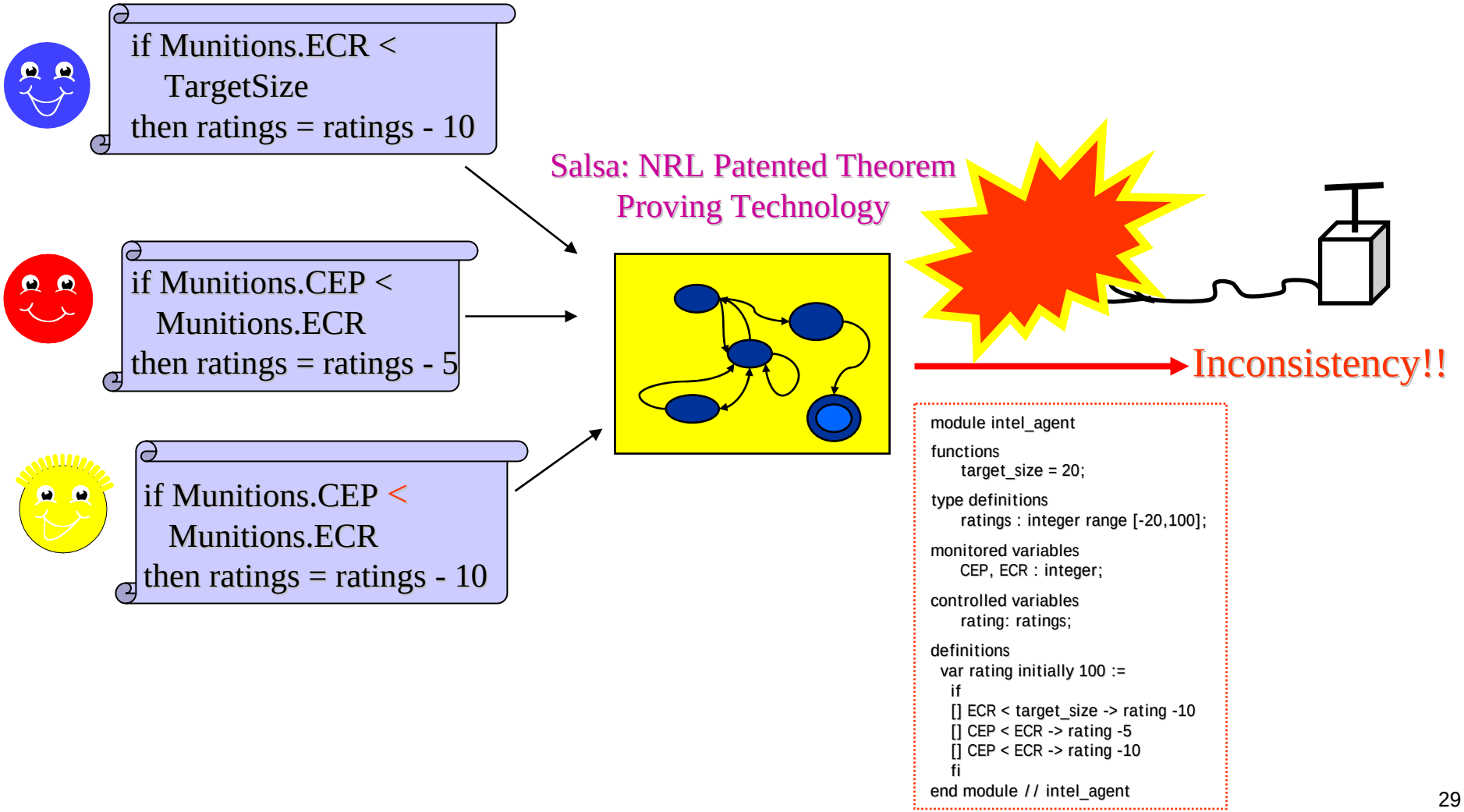
ROBUSTNESS

- Secure Operations Language (SOL)
- Agent Creation Framework
- Assurance of agent behavior

Proposed SINS Architecture

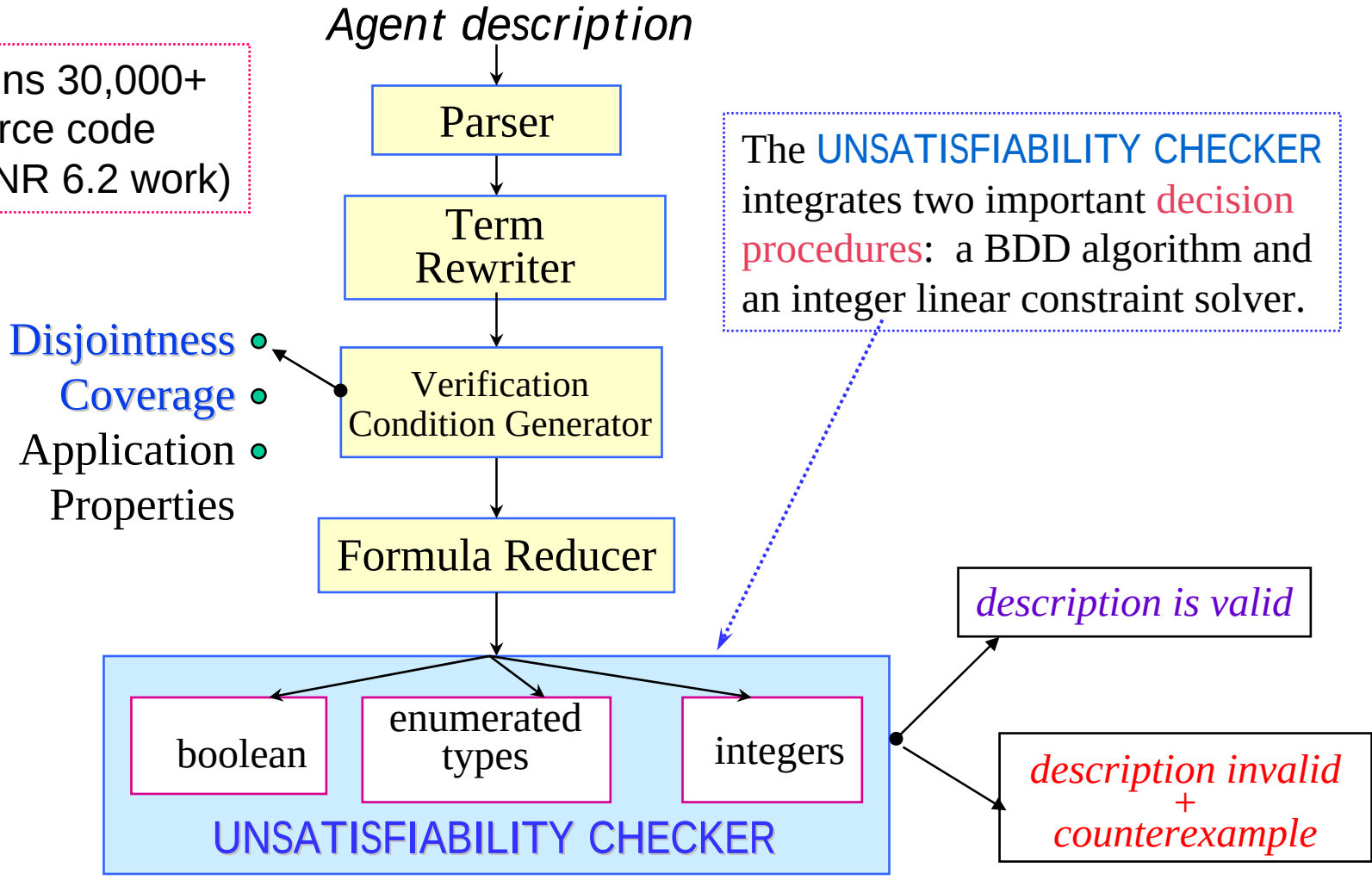


Checking Consistency of Emergent Agent Behavior



Salsa: An Automatic Invariant Checker

Salsa contains 30,000+ lines of source code (previous ONR 6.2 work)



The UNSATISFIABILITY CHECKER integrates two important decision procedures: a BDD algorithm and an integer linear constraint solver.

1. Background and Motivation
2. Our Solution
3. Design Philosophy
4. Case Studies
5. Technical Approach
6. **Major Accomplishments**
7. Transition Plans

Additional Publications

[Bha02] Bharadwaj R. "Verifiable Middleware for Secure Agent Interoperability," In Proc. Second Goddard IEEE Workshop on Formal Approaches to Agent-Based Systems (FAABS II), October 2002.

[Bha03a] Bharadwaj R. "A Framework for the Formal Analysis of Multi-Agent Systems," In Proc. Formal Approaches to Multi-Agent Systems (FAMAS) affiliated with the European Joint Conferences on Theory and Practice of Software (ETAPS 2003), Warsaw Poland, April 2003.

[Bha03b] Bharadwaj R. "Secure Middleware for Situation-Aware Naval C^2 and Combat Systems," in Proc. 9th International IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003), San Juan PR, May 2003.

[KIB03] Kim S, In P, and Bharadwaj R. "An Extended Framework for the Validation and Verification of Situation-Aware Middleware Architectures," In Proc. Ground Systems Architectures Workshop (GSAW), Manhattan Beach CA, March 2003.

[TB03] Tressler E, and Bharadwaj R. "Inter-Agent Protocol for Distributed SOL Processing," NRL Memorandum Report, In Preparation.

FY 2003 Milestones

1. SOL (Secure Operations Language)
   - Design and implementation of SOL compiler for distributed agent implementation over SSL (Secure Sockets Layer) network connections [Bha03b, KIB03].
   - Development of techniques to ensure that SOL agents are composable, consistent, safe, secure, and verifiable. References [Bha02] and [Bha03a] provide details.
2. Agent monitoring and coordination
   - Design of Inter-Agent Protocol (designated the Agent Control Protocol, or ACP) and a secondary protocol (Module Transfer Protocol, or MCP) for inter-agent communication and distributed agent deployment [TB03].
3. Determining emergent properties of multi-agent systems
   - Implementation of translators SOL2SAL and SAL2SOL as interim solution for using formal verification tool Salsa (implemented in previously funded ONR 6.2 project).

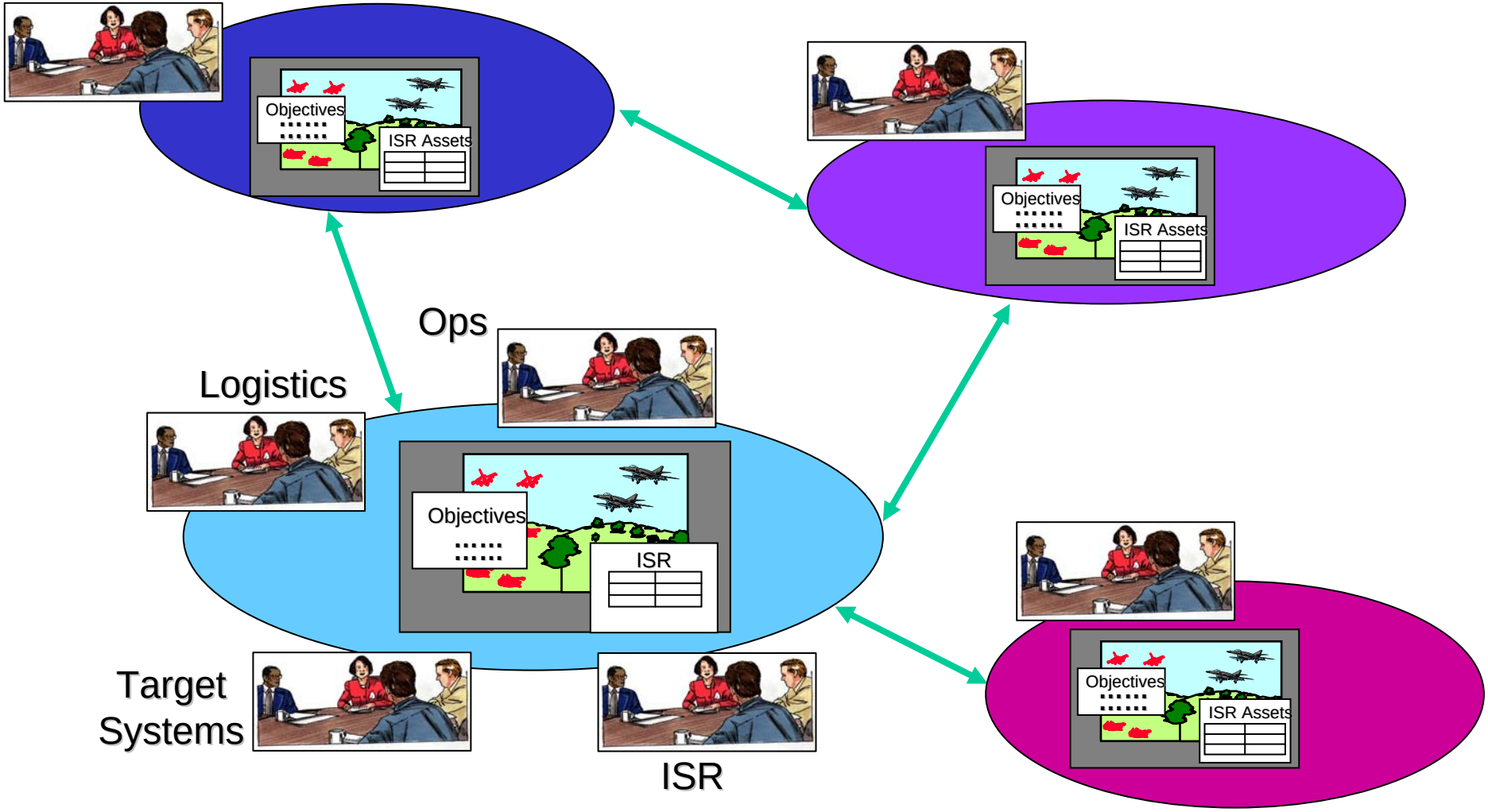
Overall Project Milestones

	FY03	FY04	FY05
• Secure Operations Language (SOL)			
- Making SOL composable, consistent, safe, secure, verifiable	♦		
- Formal proofs of application properties	o	o	♦
S Secure Infrastructure for Networked Systems (SINS)			
- Prototype Implementation	♦		
- Requirements Elicitation and Design	o	♦	
- Demonstration System		o	♦
• Agent monitoring and coordination			
- Monitoring architecture over physically distributed domains	♦		
- Selecting security protocols to enforce/maintain consistency	o	♦	
- Establishing the consistency of agent behavior and data		♦	
- Establishing that agents enforce a consistent security policy		♦	
- Obtaining a situational awareness picture for agents			♦
• Security Agents:			
- Establishing trust in security agents			♦
• Development of application-specific security agents:			
- Intrusion detection			♦
- Survivability and adaptability			♦

Key:
♦ Milestone
o Ongoing Activity

Operational Payoff:

Secure and Efficient C² for Combat Systems



Multi-Security Levels: One Role for Security Agents

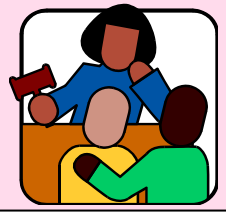
Domain A

Security agents make decisions

Sanitize information

Release policy server

optional process (e.g., remove source, fuzz image)



Enforce organization or application-specific release policy

Flow controller

Enforce flow direction

Domain B

Security agents make decisions

Receive policy server

Enforce authentication, integrity, labeling, …, policy

Security Agents ensure secure dissemination of information across domains

1. Background and Motivation
2. Our Solution
3. Design Philosophy
4. Case Studies
5. Technical Approach
6. Major Accomplishments
7. **Transition Plans**

Transition Opportunities

- Navy's Open Architecture Computing Environment
  - Aegis-equipped cruisers and destroyers
  - SSDS-equipped carriers and large deck amphibs
  - Submarines
  - DD(X) land attack destroyer
  - Littoral Combat Ship (LCS)
- UAV Swarms
- Distributed Sensor Networks

Open Architecture Characteristics

Designers have identified the following requirements:

- Portability
- Location transparency
- Loosely coupled components
  - Time and space
- Preservation of data integrity across threads, processes, computers, networks

NRL Secure Agents Middleware will provide these characteristics.

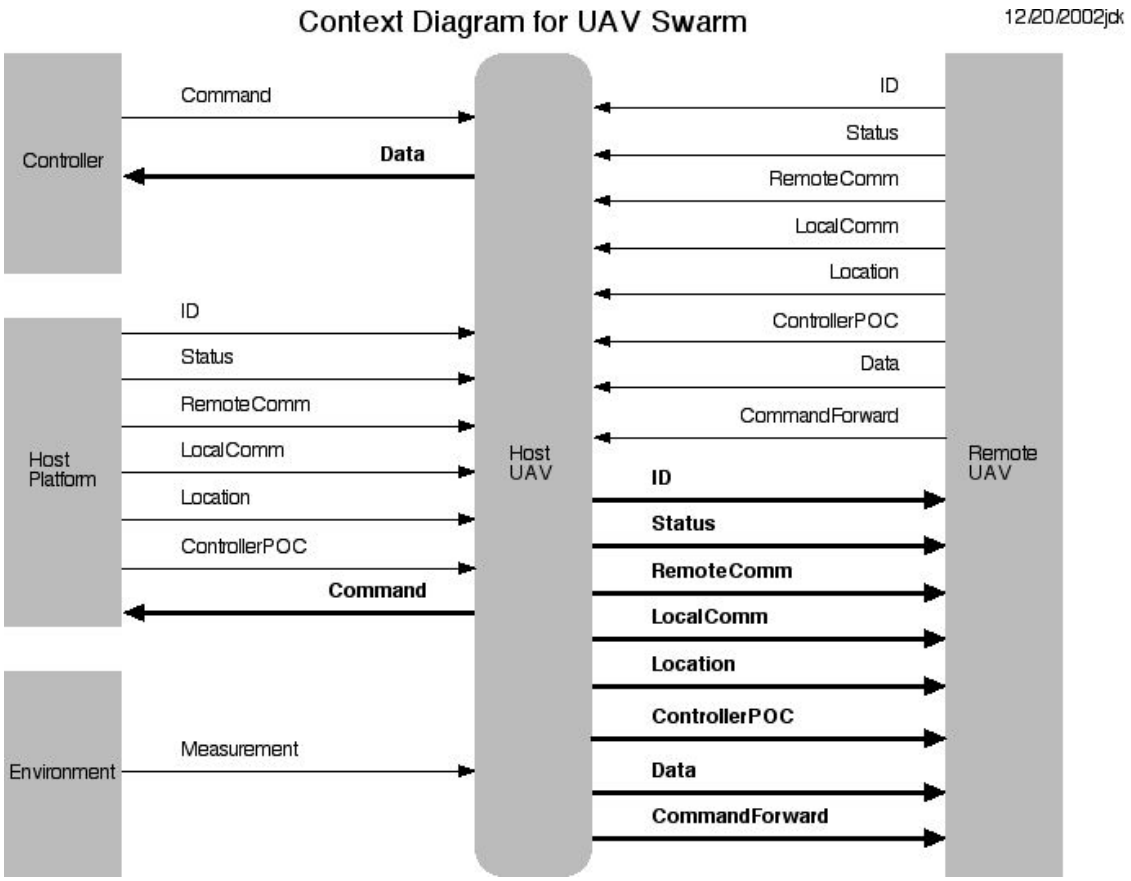
Open Architecture Challenges Addressed by SINS

We have identified the following additional challenges:

- Security
  - Malicious users
  - Malicious code
  - Confidentiality
- Impact of COTS upgrades on applications
  - Immature standards
  - 30 year lifetime
  - Vendor-specific changes
- Difficulty of finding (COTS) middleware talent
- Complexity of (COTS) middleware

How to design applications with the desired characteristics?

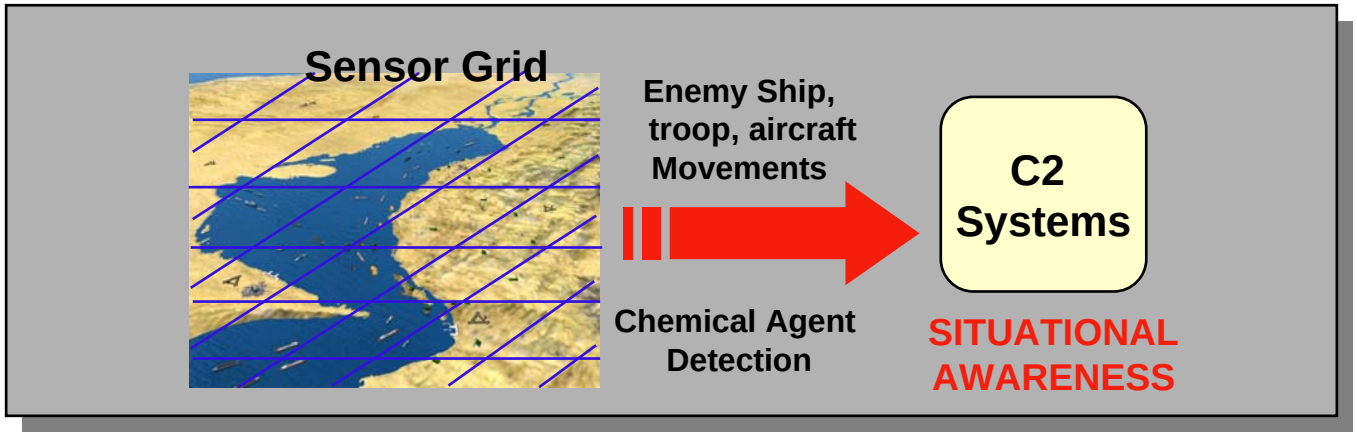
Agents for UAV Swarms



Sensor Networks

Sensor networks collect and transfer information critical to provide a complete, accurate and trusted situational awareness picture

If this information cannot be trusted, it cannot be utilized



Sensor networks are thus critical components

Their security is critical!

Sensor Network Characteristics

Sensor Attributes
- Power Constrained
- Limited Memory
- Limited Processor Capability
- Expendable

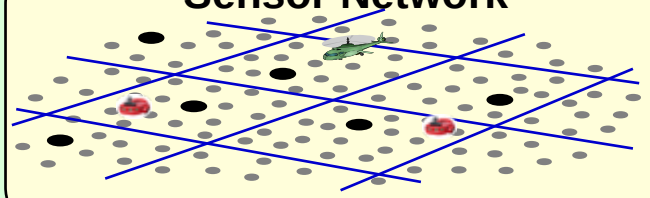
Communication Capabilities
- Wireless Interface
- Limited Bandwidth
- Limited Range

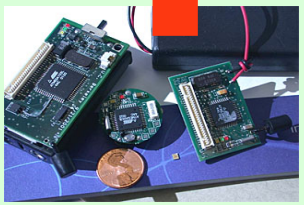
Networking
- Ad Hoc
- Self-Organizing
- Randomly Failing Nodes
- Dynamic Routing

Security Threats

Denial of Service (e.g., Jamming)
Compromise (Sensor, Network)
Injection of False Data
Spoofing

Sensor Network





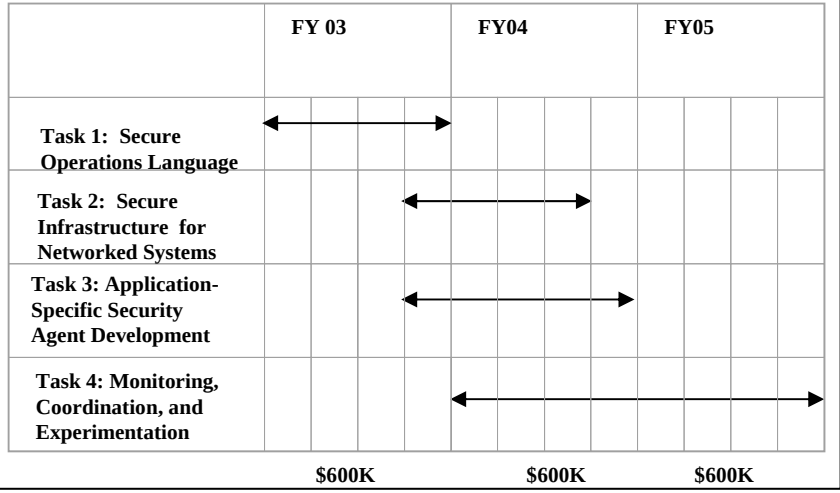
Mote (tiny, wireless) Sensor

Secure Middleware For Distributed Applications

Project Description & Technical Approach

Design and advanced prototype development of secure distributed middleware for efficient, reconfigurable, and scalable system interoperability, using the novel concept of "security agents," i.e., mini-firewalls, to ensure system integrity, efficiency and robustness. Target applications are information network situational awareness, networked C^2 for combat applications, the Open Architecture, and Unmanned Aerial Vehicle (UAV) swarms.

Project Start/Milestones/Funding



Project Objectives

Ensure secure, efficient, and robust distributed system interoperability. Additionally, reduce total ownership costs, allow quick and easy system upgrade and reconfiguration, lower the impact of COTS upgrades, and reduce compatibility problems.

Project Payoff/Impact on Naval Needs

- Networked systems that are provably secure and intrusion tolerant
- Networked systems that are flexible, reconfigurable, and survivable
- New ways of tackling *complexity*, the Achilles heel of system vulnerabilities
- Introduces a novel notion of security agents – software that polices malevolent foreign code

END